Container Registries

- An image registry needs to be part of your container plan
- More Docker Hub details including auto-build
- How Docker Store (store.docker.com) is different then Hub
- How Docker Cloud (cloud.docker.com) is different then Hub
- Use new Swarms feature in Cloud to connect Mac/Win to Swarm
- Install and use Docker Registry as private image store
- 3rd Party registry options

Docker Hub: Digging Deeper

- The most popular public image registry
- It's really Docker Registry plus lightweight image building
- Let's explore more of the features of Docker Hub
- Link GitHub/BitBucket to Hub and auto-build images on commit
- Chain image building together

Docker Store: What Is It For?

- Download Docker "Editions"
- Find certified Docker/Swarm plugins and commercial certified images

Docker Cloud: CI/CD and Server Ops

- Web based Docker Swarm creation/management
- Uses popular cloud hosters and bring-your-own-server
- Automated image building, testing, and deployment
- More advanced then what Docker Hub does for free
- Includes an image security scanning service

Running Docker Registry

- A private image registry for your network
- Part of the docker/distribution GitHub repo
- The de facto in private container registries
- Not as full featured as Hub or others, no web UI, basic auth only
- At its core: a web API and storage system, written in Go
- Storage supports local, S3/Azure/Alibaba/Google Cloud, and OpenStack Swift

Running Docker Registry Cont.

- Look in section resources for links to:
- Secure your Registry with TLS
- Storage cleanup via Garbage Collection
- Enable Hub caching via "--registry-mirror"

Run a Private Docker Registry

- Run the registry image on default port 5000
- Re-tag an existing image and push it to your new registry
- Remove that image from local cache and pull it from new registry
- Re-create registry using a bind mount and see how it stores data

Registry and Proper TLS

- "Secure by Default": Docker won't talk to registry without HTTPS
- Except, localhost (127.0.0.0/8)
- For remote self-signed TLS, enable "insecure-registry" in engine

Run a Private Docker Registry Recap

- Run the registry image
  - `docker container run -d -p 5000:5000 --name registry registry`
- Re-tag an existing image and push it to your new registry
  - `docker tag hello-world 127.0.0.1:5000/hello-world`
  - `docker push 127.0.0.1:5000/hello-world`
- Remove that image from local cache and pull it from new registry
  - `docker image remove hello-world`
  - `docker image remove 127.0.0.1:5000/hello-world`
  - `docker pull 127.0.0.1:5000/hello-world`
- Re-create registry using a bind mount and see how it stores data
  - `docker container run -d -p 5000:5000 --name registry -v $(pwd)/registry-data:/var/lib/registry registry`

Remember To Cleanup!

- No containers created in this Lecture are required for future Lectures

Private Docker Registry with Swarm

- Works the same way as localhost
- Because of Routing Mesh, all nodes can see 127.0.0.1:5000
- Remember to decide how to store images (volume driver)
- NOTE: All nodes must be able to access images
- ProTip: Use a hosted SaaS registry if possible